Anticipatory Caching in Small Cell Networks: A Transfer Learning Approach

Ejder Bastug, Mehdi Bennis, Mérouane Debbah

Anticipatory Caching in Small Cell Networks: A Transfer Learning Approach

Ejder Baştuğ$^\diamond$, Mehdi Bennis* and Mérouane Debbah$^\diamond$,
$^\diamond$Alcatel-Lucent Chair - SUPÉLEC, Gif-sur-Yvette, France
*Centre for Wireless Communications, University of Oulu, Finland
{ejder.bastug, merouane.debbah}@supelec.fr, bennis@ee.oulu.fi

I. Extended Abstract

Locally caching contents at the network edge constitutes one of the five most disruptive paradigms in 5G networks [1]. Recent results have shown that dynamic caching can significantly offload different parts of the network including the Radio access networks (RANs) and core network (CN), by smartly prefetching and storing contents closer to the end-users. In parallel to that, the era of pushing contents through the network on a *best-effort* basis ignoring who end-users are and what they are doing with their devices has dawned, calling for a truly context-aware and proactive networking paradigm [2]. As a result, edge caching has taken the recent 5G literature by storm over the last few years as evidenced in [2]–[6]. In [6], the concept of femtocaching was proposed addressing the problem of capacity-limited backhaul links by embedding base stations with high-storage units. In [2], a novel edge-centric networking paradigm was proposed in which network nodes (i.e., base station (BS) and/or user terminals (UTs)) proactively cache judiciously selected contents at the network edge. Exploiting both spatial and social caching coupled with suitable device-to-device (D2D) communication was shown to efficiently offload the backhaul traffic and enhancing the overall network performance in terms of cache-hit-rates and users' satisfaction ratios. Therein, a *proactive* caching procedure is formulated as a supervised machine learning problem and collaborative filtering (CF) techniques are used to estimate the file popularity matrix exploiting users-files correlations. Notwithstanding this fact, the file popularity matrix remains typically large and sparse in practice, rendering CF techniques sub-optimal suffering from *data sparseness* and *cold-start* problem, which are major challenges in the machine learning community [7].

In this work, we build upon the work in [2] and propose a more efficient machine learning technique, using the framework of *transfer learning (TL)* [7]. Indeed, in many real-world applications, it is expensive or even impossible to collect and label training data to build suitable prediction models. With this in mind, TL is seen as a suitable framework which allows to exploit data from other rich information sources (referred to as *source domain*) to further improve the prediction task in the *target domain*. TL has been traditionally used in data mining problems such as classification, regression and very recently in the context of CF [7].

1) Transfer Learning: In a nutshell, TL can be classified into *inductive*, *transductive* and *unsupervised* transfer learning depending on the availability of the source and target domain labels. Research issues in transfer learning boil down to: 1) what to transfer, 2) how to transfer, and 3) when to transfer. While "what to transfer" studies which part of the knowledge can be transferred across domains or tasks, "when to transfer" deals with the issue of knowing when is best to transfer the knowledge to avoid negative transfer, notably when the source and target domains become unrelated. Finally, "how to transfer" deals with knowledge extraction which needs to be transferred. A comprehensive survey of transfer learning is found in [7].

2) Contribution: The basic idea of the proposed learning approach is to alleviate the data sparsity problem encountered in most CF problems, by learning and transferring the rich contextual information (i.e., source domain), to better estimate the (large-scale) file popularity matrix in the target domain. We assume that the knowledge extracted from the source domain stems from the interaction of users accessing/sharing and recommending files within their social community via D2D (Web2.0-like). Instead of *learning from scratch* in the target domain, the nice feature of the TL approach lies in judiciously extracting collaborative social behavior information from the source domain to aid in the learning in the target domain, which will be explained in the sequel. To the best of our knowledge this is perhaps the first contribution of transfer learning in RANs.

A. Network Model

We assume that there exists an information system S_{CRP} in the source domain and an information system S_{tar} in the target domain. An illustration of the scenario is given in Fig. 1.

1) Target Domain: We consider a network deployment which consists of M_{tar} small base stations (SBSs) from the set $\mathcal{M}_{tar} = \{1, \dots, M_{tar}\}$ and N_{tar} UTs from the set $\mathcal{N}_{tar} = \{1, \dots, N_{tar}\}$. According to this setup, UTs seek certain files from a library $\mathcal{F}_{tar} = \{1, \dots, F_{tar}\}$, where each file has length of L.

We assume that every SBS is connected to the core network via a limited backhaul link with capacity C_b and every SBS has a total wireless link capacity of C_w. In order to offload the

This research has been supported by the ERC Starting Grant 305123 MORE (Advanced Mathematical Tools for Complex Network Engineering), the SHARING project under the Finland grant 128010 and the project BESTCOM.

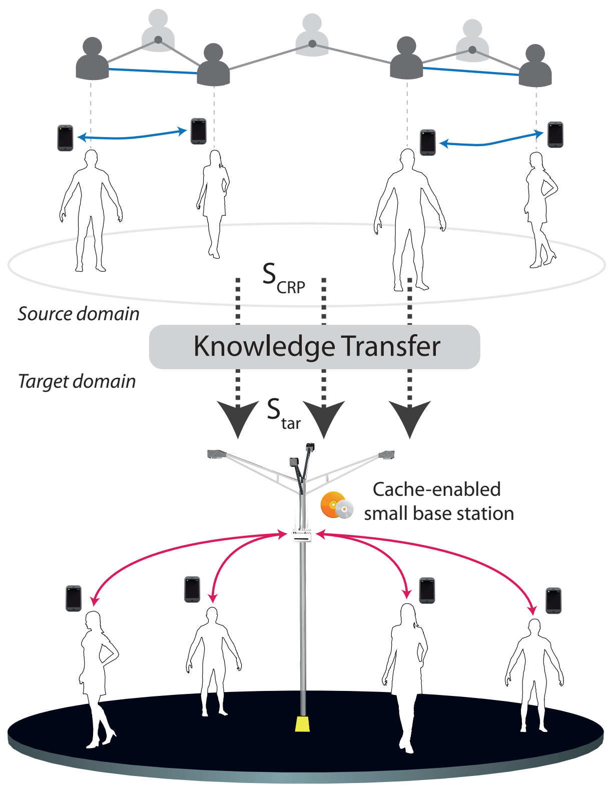

Figure 1: An illustration of the considered scenario which consists of information systems S_{CRP} and S_{tar}. The knowledge obtained from social interactions via D2D communications in source domain is transferred to the target domain.

backhaul (and satisfy users' requests more efficiently), every SBS needs to proactively fetch strategic contents from its CN and cache them at the edge. First, suppose that there exist D number of requests from the set $\mathcal{D}$ over T time slots. Suppose also a caching indicator matrix $\mathbf{\Theta} \in \{0,1\}^{M_{tar} \times F_{tar}}$, where $\theta_{m,f} = 1$ indicates that the m-th SBS stores the f-th file and $\theta_{m,f} = 0$ otherwise. Then, maximization of backhaul *offloading gain* for a fixed $\mathbf{\Theta}$ policy over T time slots can be formally written as:

$$\begin{aligned} \underset{\mathbf{\Theta}}{\text{maximize}} \quad & \frac{1}{D} \sum_{d \in \mathcal{D}} \mathbb{1}\{\theta_{n_d, f_d}\} \\ \text{subject to} \quad & \text{trace}(\mathbf{\Theta}^T \mathbf{\Theta}) \leq \frac{S}{L}, \end{aligned} \tag{1}$$

where n_d and f_d are the accessed SBS and file for request d, $\mathbb{1}\{.\}$ is the indicator function and S is the total storage capacity. In the sequel, this is referred to as the *target domain* caching formulation. Solving this problem is highly challenging due to: i) limited SBS storage capacity; ii) large number of users and library size; iii) SBSs need to track, learn and estimate user ratings given a sparse popularity matrix.

2) Source Domain: Inspired from [2], we exploit the contextual social network overlay composed of users' interactions within their social communities, referred to in the sequel as the *source domain*. The source domain considered in this work represents the behavior of users' interactions within their own social communities via D2D, modeled as a Chinese restaurant process (CRP) [2]. That is, within every social community, users sequentially request to download their sought-after content, and when a user downloads its content, the recorded hits are recorded (i.e., history). This action affects the probability that this content will be requested by others users within the same social community, where popular contents are requested more frequently and new contents less frequently.

In the target domain, the caveat of the CF-based caching policy in [2] is the fact that the file popularity matrix is largely unknown yielding slow convergence, and suffering from the *cold-start* problem. This is expected to be even more severe in settings where the number of users and files grow very large. Motivated by this fact, in this work, we propose a novel proactive caching procedure by exploiting the rich contextual information extracted from the D2D interactions via a transfer-learning procedure to more efficiently cache contents at the network edge in the target domain (i.e., higher cache-hit ratios). This caching procedure is shown to outperform classical CF-based learning methods, such as in [2].

B. Classical CF-based Learning

The classical CF-based learning is composed of a training and prediction part. In the training part, the goal is to estimate the popularity matrix $\mathbf{P}_{tar} \in \mathbb{R}^{N_{tar} \times F_{tar}}$, where every SBS builds a model based on the already available information regarding users' ratings. As given in the previous section, let $\mathcal{N}_{tar}$ and $\mathcal{F}_{tar}$ denote the set of users and files associated with N_{tar} users and F_{tar} files. In details, $\mathbf{P}_{tar}$ with entries $P_{tar,ij}$ is the (sparse) file popularity matrix in the target domain. $\mathcal{R}_{tar} = \{(i,j,r) : r = P_{tar,ij}, P_{tar,ij} \neq 0\}$ refers to the set of known user ratings. In the prediction part, in order to predict the unobserved ratings in $\mathcal{N}_{tar}$, low-rank factorization techniques are used to estimate the missing entries of $\mathbf{P}_{tar}$. The goal here is to approximate the popularity matrix $\mathbf{P}_{tar} \approx \mathbf{N}_{tar}^T \mathbf{F}_{tar}$, where the factor matrices $\mathbf{N}_{tar} \in \mathbb{R}^{k \times N_{tar}}$ and $\mathbf{F}_{tar} \in \mathbb{R}^{k \times Ftar}$ can be learned by minimizing the cost function as follows:

$$\begin{aligned} \underset{(i,j) \in \mathbf{P}_{tar}}{\text{minimize}} \quad & \sum_{(i,j) \in \mathbf{P}_{tar}} \Big(\mathbf{n}_i^T \mathbf{f}_j - P_{tar,ij} \Big)^2 + \\ & \lambda \Big(||\mathbf{N}_{tar}||_F^2 + ||\mathbf{F}_{tar}||_F^2 \Big) \end{aligned} \tag{2}$$

where the sum is over the (i,j) user/file pairs in the training set. Additionally, $\mathbf{n}_i$ and $\mathbf{f}_j$ are the i-th and j-th columns of $\mathbf{N}_{tar}$ and $\mathbf{F}_{tar}$ respectively, and $||.||_F^2$ represents Frobenius norm. In this minimization problem, the weight λ is chosen to balance between regularization and fitting training data. However, it turns out that users may rate only very few items, causing the $\mathbf{P}_{tar}$ to be extremely sparse, and thus directly minimizing (2) will suffer from severe over-fitting problems.

C. TL-based Content Caching

As alluded to earlier, exploiting and transferring the vast amount of available user-file ratings from a different-yet-related source domain can help alleviate data sparsity and solve (2) more efficiently. This is precisely the objective of this work. Formally speaking, we model the source domain S_{CRP}, which is associated with a set of N_{CRP} users and

F_{CRP} files denoted by $\mathcal{N}_{CRP}$ and $\mathcal{F}_{CRP}$. The user-file popularity matrix in the source domain is represented by matrix $\mathbf{P}_{CRP} \in \mathbb{R}^{N_{CRP} \times F_{CRP}}$ and likewise let $\mathcal{R}_{CRP} = \{(i,j,r) : r = P_{CRP,ij}, P_{CRP,ij} \neq 0\}$ denote the set of observed user ratings in the source domain. In contrast with $\mathbf{P}_{tar}$, the popularity matrix in the source domain, $\mathbf{P}_{CRP}$, contains more information which helps explore hidden patterns of user social behavior for knowledge transfer. The basic principle is to smartly "borrow" judiciously-chosen user social behavior information from S_{CRP} to better learn S_{tar}.

The transfer learning procedure from S_{CRP} to S_{tar} consists of two interrelated steps. First, an item *correspondence* needs to be established to identify similarly-rated files in both source and target domains. Second, an optimization problem is formulated combining the source and target domains for *knowledge transfer* to jointly learn the popularity matrix in the target domain $\mathbf{P}_{tar}$. In this respect, we assume that both source and target domain belong to one information system $s \in \{S_{CRP}, S_{tar}\}$, which is associated with N_s users and F_s files denoted by $\mathcal{N}_s$ and $\mathcal{F}_s$ respectively. For each system s, we observe a sparse matrix $\mathbf{P}_s$ with entries $P_{s,ij}$. Let $\mathcal{R}_s = \{(i,j,r) : r = P_{s,ij}, P_{s,ij} \neq 0\}$ denote the set of observed user ratings in each system. We refer to the set of *shared files* as $\tilde{\mathcal{F}}$. Let $\mathcal{N}^* = \mathcal{N}_{CRP} \cup \mathcal{N}_{tar}$ and $\mathcal{F}^* = \mathcal{F}_{CRP} \cup \mathcal{F}_{tar}$ denote the union of the collections of users and files, respectively, where $N^* = |\mathcal{N}^*|$ and $F^* = |\mathcal{F}^*|$ denote the total number of unique users and files in the union of both systems.

In the proposed learning approach, we model the users $\mathcal{N}^*$ and files $\mathcal{F}^*$ by a user factor matrix $\mathbf{N} \in \mathbb{R}^{k \times N^*}$ and a file factor matrix $\mathbf{F} \in \mathbb{R}^{k \times F^*}$, where the i-th and j-th columns of these matrices are represented by $\mathbf{n}_i$ and $\mathbf{f}_j$, respectively. The goal is to approximate the popularity matrix $\mathbf{P}_s \approx \mathbf{N}_s^T \mathbf{F}_s$, where the factor matrices $\mathbf{N}$ and $\mathbf{F}$ are learned by minimizing the following cost function:

$$\underset{(i,j) \in \mathbf{P}_s}{\text{minimize}} \quad \sum_s \Big(\alpha_s \sum_{(i,j) \in \mathbf{P}_s} \Big(\mathbf{n}_i^T \mathbf{f}_j - P_{s,ij} \Big)^2 \Big) + \lambda \Big(||\mathbf{N}||_F^2 + ||\mathbf{F}||_F^2 \Big) \tag{3}$$

where the α_s is the weight of each system. In doing so, $\mathbf{P}_{CRP}$ and $\mathbf{P}_{tar}$ are jointly factorized and the set of factor matrices $\mathbf{F}_{CRP}$ and $\mathbf{F}_{tar}$ become interdependent since the features of a shared file are required to be the same for knowledge sharing.

II. Numerical Results and Discussion

The datasets for numerical setup are sampled from stochastic processes and the results are obtained by averaging out 100 Monte-Carlo realizations. The evolution of the offloading gain in the target domain with respect to the storage size ratio $\left(\frac{S}{LF_{tar}}\right)$ is shown in Fig. 2. In this figure, the following caching policies are shown for comparison:

1) *Ground Truth*: The popularity matrix ($\mathbf{P}_{tar}$) is known perfectly and used for cache decision accordingly, by simply storing the most popular files for given storage size.
2) *Random caching*: Files are cached uniformly at random regardless of the popularity matrix.

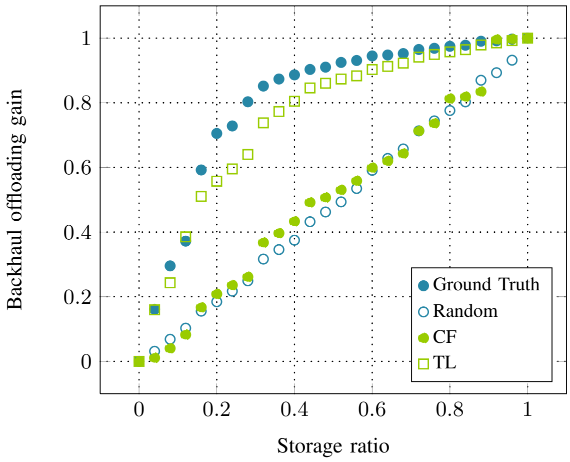

Figure 2: Evolution of the offloading gain. $M_{tar} = 1$, $N_{tar} = 32$, $F_{tar} = 32$, $L = 1$ MBit, $C_b = 1$ MBit/s, $C_w = 128$ MBit/s.

3) *CF*: $\mathbf{P}_{tar}$ is estimated via CF using a training set with 4% of rating density.
4) *TL*: $\mathbf{P}_{tar}$ and $\mathbf{P}_{CRP}$ are jointly factorized via TL using perfect correspondence and 12% of rating density in the training set.

It can be seen that the CF method is not able to approximate the ground truth well, thus, yielding poor offloading gains similar to random caching. On the other hand, the joint estimation done by TL improves the caching performance, approaching the offloading gains of ground truth.

III. Conclusions

We showed the benefits of TL approach in a scenario where cache-enabled SBSs is giving low performance due to the poor estimation of CF. Numerical results showed that offloading gains can be improved by transferring the knowledge from source domain to target domain using TL approaches. An interesting future work would be investigating the impact of various system parameters (besides storage size) where the gains can be different depending on the numerical setting.

References

[1] F. Boccardi, R. Heath, A. Lozano, T. Marzetta, and P. Popovski, “Five disruptive technology directions for 5g,” *Communications Magazine, IEEE*, vol. 52, no. 2, pp. 74–80, February 2014.

[2] E. Baştuğ, M. Bennis, and M. Debbah, “Living on the Edge: The role of Proactive Caching in 5G Wireless Networks,” *IEEE Communications Magazine*, In Press, August, 2014.

[3] E. Baştuğ, J.-L. Guénégo, and M. Debbah, “Proactive small cell networks,” in *20th International Conference on Telecommunications (ICT)*, Casablanca, Morocco, 05/2013 2013.

[4] K. Poularakis, G. Iosifidis, V. Sourlas, and L. Tassiulas, “Multicast-aware caching for small cell networks,” in *IEEE Wireless Communications and Networking Conference (WCNC 2014)*, Istanbul, Turkey, April 2014.

[5] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. Leung, “Cache in the air: exploiting content caching and delivery techniques for 5g systems,” *Communications Magazine, IEEE*, vol. 52, no. 2, pp. 131–139, February 2014.

[6] N. Golrezaei, K. Shanmugam, A. Dimakis, A. Molisch, and G. Caire, “Femtocaching: Wireless video content delivery through distributed caching helpers,” in *INFOCOM, 2012 Proceedings IEEE*, March 2012, pp. 1107–1115.

[7] S. J. Pan and Q. Yang, “A survey on transfer learning,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, no. 10, pp. 1345–1359, October 2010.